Stochastic and Spatial Equivalences for PALOMA

Citation for published version:
Piho, P & Hillston, J 2016, Stochastic and Spatial Equivalences for PALOMA. in *Proceedings of the Workshop on FORmal methods for the quantitative Evaluation of Collective Adaptive SysTems (FORECAST 2016).* Electronic Proceedings in Theoretical Computer Science, vol. 217, Open Publishing Association, pp. 69-80, Formal methods for the quantitative Evaluation of Collective Adaptive Systems, Vienna, Austria, 8/07/16. https://doi.org/10.4204/EPTCS.217.9

Digital Object Identifier (DOI):
10.4204/EPTCS.217.9

Link:
Link to publication record in Edinburgh Research Explorer

Document Version:
Peer reviewed version

Published In:
Proceedings of the Workshop on FORmal methods for the quantitative Evaluation of Collective Adaptive SysTems (FORECAST 2016)

Stochastic and Spatial Equivalences for PALOMA

Paul Piho
LFCS, School of Informatics,
University of Edinburgh
paul.piho@ed.ac.uk

Jane Hillston
LFCS, School of Informatics,
University of Edinburgh
jane.hillston@ed.ac.uk

We concentrate our study on a recent process algebra – PALOMA – intended to capture interactions between spatially distributed agents, for example in collective adaptive systems. New agent-based semantic rules for deriving the underlying continuous time Markov chain are given in terms of State to Function Labelled Transition Systems. Furthermore we define a bisimulation with respect to an isometric transformation of space allowing us to compare PALOMA models with respect to their relative rather than absolute locations.

1 Introduction

PALOMA (Process Algebra for Located Markovian Agents) [**?**, **?**] is a novel stochastic process algebra which captures the behaviour of agents who are distributed in space and whose interactions are affected by their relative positions. This model can be thought to capture many modern systems where, for example, the range of communication may be limited for devices using wireless communication technologies or some areas may be known "dead zones" from which no communication is possible. In this paper we consider what it means for two agents to be equivalent, taking into consideration both their behaviour and their location, and develop the formal underpinnings to allow such equivalence to be rigorously studied.

The notion of Markovian bisimulation has become standard for stochastic process algebras, but as we will discuss, applied naively this approach to equivalence checking is too strong, leaving little opportunity for a notion of equivalence that is not isomorphism. Instead here we consider equivalence of a component within the context of a given system. This supports the idea of being able to substitute one component, perhaps with a more efficient implementation, for another within a given system even though they may not exhibit exactly the same behaviour in arbitrary contexts. Similarly, when we come to consider the spatial aspects of behaviour our notion of equivalence aims to capture the relative positions of components, rather than their absolute locations.

In this brief paper we aim to give the intuition and ideas behind our bisimulation, without giving all the definitions. The rest of the paper is structured as follows. In Section 2 we give a brief introduction to the PALOMA modelling language, while the semantics of the language is outlined in Section 3. In Section 4 we discuss a notion of equivalence based on equivalent relative positions and behaviours. We present our conclusions and discuss further work in Section 5.

2 PALOMA language

In this section we give a brief introduction to PALOMA; the interested reader is referred to [**?**, **?**] for more details. The spatial distribution of agents is a key feature of PALOMA models and we assume that there exists a finite set of locations, *Loc* and all agent expressions in PALOMA are parameterised by a location $\ell \in Loc$, indicating the current location of the agent.

To appear in EPTCS.

The grammar of the language is as follows:

$$\pi ::= !!(\alpha,r)@\mathbf{IR}\{\overrightarrow{\ell}\} \;\mid\; ??(\alpha,p)@\mathbf{Wt}\{w\} \;\mid\; !(\alpha,r)@\mathbf{IR}\{\overrightarrow{\ell}\} \;\mid\; ?(\alpha,p)@\mathbf{Pr}\{q\} \;\mid\; (\alpha,r)$$
$$S(\ell) ::= \pi.S'(\ell') \;\mid\; S_1(\ell)+S_2(\ell) \;\mid\; C$$
$$P ::= S(\ell) \;\mid\; P \parallel P$$

The two-level grammar, defines individual agents $S(\ell)$, whose behaviours are specified by the actions they can undertake, with possible alternatives, and model components P, which are comprised of parallel compositions of agents. The behaviour of individual agents are given by actions of five distinct types:

Unicast output $!!(\alpha,r)@\mathbf{IR}\{\overrightarrow{\ell}\}$**:** Unicast is for point-to-point communication between a pair of agents and is included in the language to model contention for resources in systems. Each unicast output message has a label, α, and a rate r, that determines the rate at which the output is performed. The message is sent to locations specified by the set $\overrightarrow{\ell} \in 2^{Loc}$ interpreted as the *influence range*. Any agent located within that range, which enables the corresponding α-labelled unicast input action, is eligible to receive the action — that is, the label α is used to identify agents that can communicate with each other. Unicast actions are *blocking* meaning that the sending agent can only proceed when there is a eligible receiver.

Unicast input $??(\alpha,p)@\mathbf{Wt}\{w\}$**:** Each eligible receiver of a unicast message α must be located within the specified influence range, and each will have an associated *weight* w. The weights are used to define a probability distribution over the eligible receivers, i.e. if there are i potential receivers, each with weight w_i and $W = \sum_i w_i$ then the probability that the jth agent receives the message is w_j/W. Once the message is received the receiving agent may or may not act on the message (reflecting message failure, corruption etc.) with the specified probability p i.e. with probability $1-p$ the agent will not act on the message received. If this occurs the message is lost — it is not the case that it is subsequently assigned to one of the other eligible receivers.

Broadcast output $!(\alpha,r)@\mathbf{IR}\{\overrightarrow{\ell}\}$**:** As its name suggests, a broadcast action allows its sender to influence multiple other agents. As with the unicast output action, a broadcast output message labelled α is sent with a specified influence range $\overrightarrow{\ell}$ and at a specified rate r. *All* agents with broadcast input prefix on label α located within that range may receive the message. Moreover the output proceeds regardless of whether there are any eligible receivers so broadcast output is non-blocking for the sender.

Broadcast input $?(\alpha,p)@\mathbf{Pr}\{q\}$**:** Each eligible receiver of a broadcast message α must be located within the specified input range. Each such agent has a likelihood of receiving the message, recorded in the probability q. For example, agents closer to the sender may be more likely to receive the message. Each agent independently decides whether the broadcast is received or not (Bernoulli trials). As with unicast input, the receiving agent may or may not act on the message with the specified probability p i.e. with probability $1-p$ the agent will not act on the message received.

Spontaneous action (α,r)**:** These actions do not represent a communication but rather an individual action by the agent which may change the state of the agent, for example, its location. These can also be thought of as broadcast output actions whose influence range is the empty set.

All rates are assumed to be parameters of exponential distributions, meaning that the underlying stochastic model of a PALOMA model is a continuous time Markov chain (CTMC).

Example 2.1. Consider agents *Transmitter* and *Receiver* such that

$$\begin{aligned} \mathit{Receiver}(\ell) &:= ??(\mathit{message}, p)@\,\mathbf{Wt}\{v\}.\mathit{Receiver}(\ell) \\ \mathit{Transmitter}(\ell) &:= !!(\mathit{message}, r)@\,\mathbf{IR}\{\vec{\ell}\}.\mathit{Transmitter}(\ell) \end{aligned}$$

where ℓ denotes the current location of the agent and $\vec{\ell}$ denotes a set of locations in the range of the unicast message emitted by action *message*. In a system where no agent sends a *message* agent *Receiver* does not perform any action. On the other hand if there is a component, say *Tranmitter*, that outputs a *message* and the location of *Receiver* is in the influence range of the message then *Receiver* performs *message* with a rate dependent on the rate at which *Transmitter* unicasts *message* and the probability that *Receiver* receives it. Similarly, if the component *Transmitter* does not have a recipient for the *message*, it remains blocked and never performs an action.

2.1 Conditional exit rates and probabilities

Notions of equivalence in process algebras, such as bisimulation [**?**], are typically based on the idea of a pair of agents each being able to match the behaviour of the other. In the case of stochastic process algebras such as PEPA, not only the type of action but also the rates at which they occur must in some sense be matched [**?**]. In order to make similar definitions for PALOMA we need to define some auxiliary functions which, given a syntactic expression, extract information about the rates and probabilities which may be exhibited by the term. Space limitations do not allow us to present all of them here, but we present those for unicast, which is the most involved case, to give the reader an impression of how we proceed.

Denote the set of all sequential components of PALOMA parametrised by their location by $\mathscr{C}_S$ and the set of model components by $\mathscr{C}$. Let the set of action labels be defined as *Lab* and the set of action types as $\mathit{Type} = \{!!, ??, !, ?, \cdot\}$, where the interpretation of the symbols is clear, corresponding to the action types discussed above. Let *Act* denote the set of all actions. The actions in the set $\mathit{Act} = \mathit{Type} \times \mathit{Lab}$ are defined by their label and their type. Let $\mathscr{A}$ be the set of all syntactically defined actions. Define the function $\Pi_{Act} : \mathscr{A} \to \mathit{Act}$ as a projection returning the label of the action with its type, e.g. $\Pi_{Act}(??(\alpha, p)@\,\mathbf{Wt}\{v\}) = ??\alpha$. Similarly define the projection $\Pi_{Lab} : \mathit{Act} \to \mathit{Lab}$ returning just the label of the action and the function $\Pi_{Type} : \mathit{Act} \to \mathit{Type}$ returning the type of an action.

Denote by Π_{Loc} the function returning the set of locations spanned by a model component.

$$\Pi_{Loc}(S_1(\ell_1) \parallel \cdots \parallel S_n(\ell_n)) = \bigcup_{i=1}^{n} \{\ell_i\}$$

Note that in the case of sequential components Π_{Loc} will result in a singleton set — the location of the sequential component.

Suppose $\mathit{Sys} = S_1(\ell_1) \parallel \cdots \parallel S_n(\ell_n) \in \mathscr{C}$ for $n \in \mathbb{N}^+$. Let the function seq return the set of all sequential components of *Sys* in a set of locations *L*.

$$\mathrm{seq}(\mathit{Sys}, L) = \{S_i(\ell_i) \mid \Pi_{Loc}(S_i(\ell_i)) \in L\}$$

2.1.1 Context unaware definitions

When we consider a PALOMA component in isolation we can use the syntax to find the potential rate, weight or probability associated with this component and a given action. Similar functions are defined for each form of prefix. From the point of view of the originator of a unicast action, the important measure is the rate at which the action is preformed.

Definition 2.1. For all $\alpha \in Lab$, $a \in \mathscr{A}$, $\vec{\ell} \in 2^{Loc}$, and $S \in \mathscr{C}_S$ define the function $s^{!!}_{\alpha}$ returning the rate of a unicast output action labelled α as follows.

$$\begin{aligned}
s^{!!}_{\alpha}\left(!!(\beta,r)@\,\mathbf{IR}\{\vec{\ell}\}.S(\ell)\right) &= \begin{cases} r & \text{for } \alpha=\beta \\ 0 & \text{otherwise} \end{cases} \\
s^{!!}_{\alpha}\left(a.S(\ell)\right) &= \;0 \quad \text{if } \Pi_{Type}(a) \neq !! \\
s^{!!}_{\alpha}\left(S_1(\ell)+S_2(\ell)\right) &= \;s^{!!}_{\alpha}(S_1(\ell)) + s^{!!}_{\alpha}(S_2(\ell))
\end{aligned}$$

Example 2.2. Consider the following components

$$\begin{aligned}
Tester(\ell_0) &:= (message, r).Tester(\ell_0) \\
Transmitter(\ell_0) &:= !!(message, r).Transmitter(\ell_0) \\
Receiver(\ell_1) &:= ??(message, p)@\,\mathbf{Wt}\{v\}.Receiver(\ell_1)
\end{aligned}$$

Based on these definitions we can find:

$$s^{!!}_{message}(Tester(\ell_0) + Transmitter(\ell_0)) = 0 + r = r \qquad\qquad s^{!!}_{message}(Receiver(\ell_1)) = 0$$

The rest of the context unaware definitions are given in a similar vein and just extract necessary syntactic information from the component definitions. Specifically we define the following functions:

Unicast influence range $\Pi_{UniIR}(S,\alpha)$: Given that S has a unicast output prefix with label α, the function returns the influence range of unicast message α defined in the prefix. Otherwise, the function returns the empty set $\emptyset$.

Weight function $w_\alpha(S)$: For a sequential component S the function $w_\alpha(S)$ is defined similarly to $s^{!!}_{\alpha}$ with base case $w_\alpha\left(??(\alpha, p)@\,\mathbf{Wt}\{w\}.S\right) = w$. In addition we define the weight function over parallel compositions and sets of sequential components by summing over the weights for each sequential component in the parallel composition or set.

Probability function $p^{??}_{\alpha}(S)$: This is again similar to $s^{!!}_{\alpha}$ with base case $p^{??}_{\alpha}\left(??(\alpha, p)@\,\mathbf{Wt}\{w\}.S\right) = p$.

Example 2.3. Consider the following sequential components.

$$\begin{aligned}
Transmitter(\ell_0) &:= !!(message, r)@\,\mathbf{IR}\{\vec{\ell}\}.Transmitter(\ell_0) \\
Receiver1(\ell_1) &:= ??(message, p)@\,\mathbf{Wt}\{w_{r1}\}.Receiver1(\ell_1) \\
Receiver2(\ell_2) &:= ??(message, q)@\,\mathbf{Wt}\{w_{r2}\}.Receiver2(\ell_2)
\end{aligned}$$

For the system given by $Sys = Transmitter(\ell_0) \parallel Receiver1(\ell_1) \parallel Receiver2(\ell_2)$ the weight for receiving a unicast message *message* is calculated as

$$w_{message}(Sys) = w_{message}(Transmitter(\ell_0) \parallel Receiver1(\ell_1) \parallel Receiver2(\ell_2)) = w_{r1} + w_{r2}$$

2.2 Context-aware conditional exit rates

Unfortunately the syntactic information alone is not sufficient to determine the rate at which an action will be witnessed in a PALOMA system. The spatial aspect, as captured by the influence range, plays an important role in determining both which actions are possible and potentially their rates and probabilities. Thus we also define some context-dependent functions.

Definition 2.2. Let α be an action label in *Lab*. Define the rate at which the component $S(\ell) \in \mathscr{C}_S$ is capable of unicasting a message labelled α to a location ℓ' as follows:

$$\begin{aligned} u_\alpha(\ell', !!(\beta, r)@\mathbf{IR}\{\overrightarrow{\ell}\}.S(\ell)) &= \begin{cases} s^{!!}_\alpha(!!(\beta, r)@\mathbf{IR}\{\overrightarrow{\ell}\}.S(\ell)) & \text{if } \ell' \in \Pi_{UniIR}(S(\ell), \alpha) \text{ and } \alpha = \beta \\ 0 & \text{otherwise} \end{cases} \\ u_\alpha(\ell', S_1(\ell) + S_2(\ell)) &= u_\alpha(\ell', S_1(\ell)) + u_\alpha(\ell', S_2(\ell)) \end{aligned}$$

Definition 2.3. Suppose $P = S_1(\ell_1) \parallel \cdots \parallel S_n(\ell_n) \in \mathscr{C}$ for $n \in \mathbb{N}^+$ is a model component with $S_i(\ell_i) \in \mathscr{C}_S$ for all $1 \le i \le n$. Let *Sys* be any other system serving as context. Let $u_\alpha(\ell, Sys, P)$ be the rate at which a model component P unicasts a message labelled α to location ℓ in the context of *Sys*, defined as

$$u_\alpha(\ell, Sys, P) = \sum_{S \in \mathrm{seq}(P)} u_\alpha(\ell, S) \times \mathbb{1}_{>0}\{w_\alpha(\mathrm{seq}(Sys \parallel P, \Pi_{UniIR}(S, \alpha)))\}$$

$$\text{where } \mathbb{1}_{>0}(x) = \begin{cases} 1 & x > 0 \\ 0 & \text{otherwise} \end{cases}$$

For each sequential component S of P we calculate the total weight over the components in the influence range of S. The indicator function $\mathbb{1}_{>0}$ is set to 1 if this weight is greater than 0 — meaning there are eligible receivers in the influence range. The rate at which P unicasts a message α to location ℓ is then defined as the sum of rates at which each sequential component S of P is *capable* of said unicast multiplied by the indicator function ensuring that the blocking nature of unicast is taken into account.

The next definition deals with determining the probability of a sequential component receiving the unicast message.

Definition 2.4. Let $S_1(\ell)$ and $S_2(\ell')$ be sequential components and $Sys \in \mathscr{C}$ any model component. Suppose $!!(\alpha, r)@\mathbf{IR}\{\overrightarrow{\ell}\}.S'_2(\ell'')$ is a prefix guarded term in the expression of $S_2(\ell')$. Then we define the probability of $S_1(\ell)$ receiving a unicast message with label α from $S_2(\ell')$, when composed in parallel with *Sys* and $S_2(\ell')$, to be:

$$p_\alpha(S_1(\ell), Sys, S_2(\ell')) = \begin{cases} \frac{w_\alpha(S_1(\ell))}{w_\alpha(\mathrm{seq}(Sys \| S_1(\ell), \overrightarrow{\ell}))} & \text{if } \ell \in \overrightarrow{\ell} \\ 0 & \text{otherwise} \end{cases}$$

Once similar definitions have been defined for broadcast and spontaneous actions we are in a position to define the context-aware exit rate.

Definition 2.5. Let $Sys \in \mathscr{C}$ be a system in which the component $S \in \mathscr{C}_S$ appears. Let $a \in \mathscr{A}$ be any action with label α. Define the *context-aware exit rate* R for agents by the following:

$$R_a(Sys, S(\ell_0)) = \begin{cases} s_\alpha(S(\ell_0)) & \text{if } \Pi_{Type}(a) = \cdot \\ b_\alpha(S(\ell_0)) & \text{if } \Pi_{Type}(a) = ! \\ b_\alpha(\ell_0, Sys) p^?_\alpha(S(\ell_0)) & \text{if } \Pi_{Type}(a) = ? \\ \max\limits_{\ell \in \Pi_{Loc}(Sys)} \{u_\alpha(\ell, Sys, S(\ell_0))\} & \text{if } \Pi_{Type}(a) = !! \\ \sum\limits_{T \in Seq(Sys)} u_\alpha(\ell_0, Sys, T) p_\alpha(S(\ell_0), Sys, T) p^{??}_\alpha(S(\ell_0)) & \text{if } \Pi_{Type}(a) = ?? \end{cases}$$

Now consider a model component $P = S_1(\ell_1) \parallel \cdots \parallel S_n(\ell_n)$ with $S_i(\ell_i) \in \mathscr{C}_S$ for all $1 \le i \le n$ ($n \in \mathbb{N}$) and suppose it is a part of the system *Sys*. Then define

$$R_a(Sys, P) = \sum_{i=1}^{n} R_a(Sys \parallel (P \setminus S_i(\ell_i)), S_i(\ell_i))$$

where $P \setminus S_i(\ell_i)$ denotes the model component P with $S_i(\ell_i)$ removed.

Finally we define the rate at which action $a \in Act$ is performed over a set of locations.

Definition 2.6. Consider a model component $P = S_1(\ell_1) \parallel \cdots \parallel S_n(\ell_n)$ with $S_i(\ell_i) \in \mathscr{C}_S$ for all $1 \leq i \leq n$ ($n \in \mathbb{N}$) and suppose it is a part of the system *Sys*. Let L be a set of locations of interest. We define $R_a(L, Sys, P)$, the rate at which action a is performed by P in locations L, within the context of system *Sys* to be:

$$R_a(L, Sys, P) = \sum_{S \in \mathrm{seq}(P,L)} R_a(Sys \parallel (P \setminus S), S)$$

3 Semantics

The definition of the semantics of PALOMA will proceed in the FuTS (State to Function Labelled Transition Systems) framework as presented in [**?**]. In general, the transition rules in FuTSs are given as triplets $s \overset{\lambda}{\rightarrowtail} f$ where s denotes a source state, λ the label of the transition and f the continuation function associating a value of suitable type to each state s'. The shorthand $[s_1 \mapsto v_1, \cdots, s_n \mapsto v_n]$ is used to denote a function f such that $f(s_i) = v_i$ for $i = 1, \cdots n$. This kind of functional treatment of transition rules is going to allow us to give more concise definitions of semantic rules as many possible branches of model evolutions can be captured within a single rule.

In the case of PALOMA semantics we are going to define the set of states as the set of all model components $\mathscr{C}$. For convenience, the treatment of semantic rules is split into two steps where the following types of transition relations are considered separately:

Capability relation Denoted by $s \overset{\lambda}{\rightarrowtail}_c f$ where $f : \mathscr{C} \rightarrow [0,1]$. The aim is to describe actions that a defined model component is capable of and introduce probabilities for all possible states resulting from the said action firing. For example, a component including a prefix for unicast input will be capable of the unicast input action firing with some probability dependent on the context. The function f will assign a probability for possible continuation states.

Stochastic relation Denoted by $s \overset{\lambda}{\rightarrowtail}_s f$ where $f : \mathscr{C} \rightarrow \mathbb{R}^{\geq 0}$. These rules are used to generate the CTMC and thus need to assign rates to each available transition.

As mentioned in Section 2, the calculation of rates of actions for each component depend the system they appear in (a PALOMA model component) and thus we use *Sys* as a place-holder for any such PALOMA model component serving as context. In the following, we use $P_1 \equiv P_2$ to denote that model components P_1 and P_2 are syntactically equivalent.

3.1 Capability relations

The only capability relations of interest here are ones for broadcast and unicast input actions as these are the only ones that can either succeed or fail depending on the rest of the context system *Sys*.

The labels λ_c, of the FuTSs rules are given by the following grammar where $\alpha \in \mathsf{Lab}$ denotes the action labels:

$$\begin{array}{rll} \lambda_c ::= & (?\alpha, \vec{\ell}, Sys) & \text{Broadcast input} \\ & \mid (??\alpha, \vec{\ell}, Sys) & \text{Unicast input} \end{array}$$

BrIn $\quad ?(\alpha,p)@\mathbf{Pr}\{q\}.S \xrightarrow{(?\alpha,\overrightarrow{\ell},Sys)}_c f \quad$ if $\Pi_{Loc}(?(\alpha,p)@\mathbf{Pr}\{q\}.S) \in \overrightarrow{\ell}$

$$f(s) = \begin{cases} pq & \text{if } s \equiv S \\ 1-pq & \text{if } s \equiv ?(\alpha,p)@\mathbf{Pr}\{q\}.S \\ 0 & \text{otherwise} \end{cases}$$

UniIn $\quad ??(\alpha,p)@\mathbf{Wt}\{w\}.S \xrightarrow{(??\alpha,\overrightarrow{\ell},Sys)}_c f \quad$ if $\Pi_{Loc}(??(\alpha,p)@\mathbf{Wt}\{w\}.S) \in \overrightarrow{\ell}$

$$f(s) = \begin{cases} \dfrac{wp}{w_\alpha(Seq)} & \text{if } s \equiv S \\ \dfrac{w(1-p)}{w_\alpha(Seq)} & \text{if } s \equiv ??(\alpha,p)@\mathbf{Wt}\{w\}.S \\ 0 & \text{otherwise} \end{cases}$$

where $Seq = \text{seq}(Sys, \overrightarrow{\ell})$

BrSystem $\quad \dfrac{P_1 \xrightarrow{(?\alpha,\overrightarrow{\ell},Sys)}_c f_1 \quad P_2 \xrightarrow{(?\alpha,\overrightarrow{\ell},Sys)}_c f_2}{P_1 \parallel P_2 \xrightarrow{(?\alpha,\overrightarrow{\ell},Sys)}_c g}$

$$g(s) = \begin{cases} f_1(P_1')f_2(P_2') & \text{if } s \equiv P_1' \parallel P_2' \\ 0 & \text{otherwise} \end{cases}$$

ParllelUniIn $\quad \dfrac{S_1 \xrightarrow{(??\alpha,\overrightarrow{\ell},Sys)}_c f_1 \quad S_2 \xrightarrow{(??\alpha,\overrightarrow{\ell},Sys)}_c f_2}{S_1 \parallel S_2 \xrightarrow{(??\alpha,\overrightarrow{\ell},Sys)}_c g}$

$$g(s) = \begin{cases} f_1(S_1') & \text{if } s \equiv S_1' \parallel S_2 \\ f_2(S_2') & \text{if } s \equiv S_1 \parallel S_2' \\ 0 & \text{otherwise} \end{cases}$$

Choice $\quad \dfrac{P_1 \xrightarrow{\lambda_c}_c f}{P_1 + P_2 \xrightarrow{\lambda_c}_c f} \qquad \dfrac{P_2 \xrightarrow{\lambda_c}_c f}{P_1 + P_2 \xrightarrow{\lambda_c}_c f}$

Constant $\quad \dfrac{P \xrightarrow{\lambda_c}_c f \qquad X := P}{X \xrightarrow{\lambda_c}_c f}$

Figure 1: Capability rules for communication

The semantic rules given in Figure 1 use the definitions from Section 2 to extract necessary information from the syntactic definitions of components.

The rules BrIn and UniIn are the primitive rules describing the capability of sequential components to perform a broadcast or unicast input action, respectively, given the set of locations $\overrightarrow{\ell}$ denoting the influence range of the message and a context system *Sys*. In both cases the function f, which is defined over all states, gives the probability of a transition to each state given the action has fired. For BrIn the calculation only depends on the parameters p and q given explicitly in the syntactic definition of the component. For UniIn the likelihood of the component receiving the message, $\frac{w}{w_\alpha(Sys)}$, is calculated on the basis that there may be many eligible receivers of the given message in *Sys*.

The rule BrSystem is used to deal with parallel compositions of model components that can act as broadcast message receivers. Note that the outcomes of all the broadcast input actions in a system are independent of each other. Thus the probability of $P_1 \parallel P_2$ transitioning to $P_1' \parallel P_2'$ due to a broadcast input action is the product of the probabilities of P_1 and P_2 respectively making the corresponding transitions.

For unicast input actions the rule ParllelUniIn is just saying that no two components can perform the unicast input on the same label simultaneously.

3.2 Stochastic relations

Firstly we need to define a set of labels for stochastic relations. It will be necessary to carry around the set of locations $\overrightarrow{\ell}$ in the labels to distinguish between actions having the same label and type but affecting a different set of components due to their influence range. In addition, including the system *Sys* in the labels ensures that the communication rules are only applied to components in the same system.

The set of labels for stochastic relations is thus defined as follows:

$$\begin{array}{rll} \lambda_s ::= & (\alpha,\ \emptyset,\ Sys) & \text{Spontaneous action} \\ & \mid (!\alpha,\ \vec{\ell}, Sys) & \text{Broadcast communication} \\ & \mid (!!\alpha, \vec{\ell}, Sys) & \text{Unicast communication} \end{array}$$

The stochastic rules are summarised in Figure 2. Firstly we have rules Br, Uni and SpAct that just define the primitive rules for all spontaneous actions and give the rates at which the defined transitions can happen. For the rule Uni the side-condition is needed to ensure that there are eligible receivers available in the system.

The rules BrCombo and UniPair are to combine the capability rules with stochastic rules to give rates of system state transitions that are induced by broadcast or unicast message passing. BrCombo takes as premise the existence of components S and P such that S can perform the broadcast communication action defined by stochastic relations and P is capable of broadcast input. The rate at which the parallel composition $S \parallel P$ reaches the next state $S' \parallel P'$ is given by the function $f \otimes g$ which is defined as the product of f applied to the S' and g applied to P. The unicast case is treated similarly.

Br $\quad !(\alpha, r)@\mathbf{IR}\{\vec{\ell}\}.S \xrightarrow{(!\alpha,\vec{\ell},Sys)}_s [S \mapsto r]$

Uni $\quad !!(\alpha, r)@\mathbf{IR}\{\vec{\ell}\}.S \xrightarrow{(!!\alpha,\vec{\ell},Sys)}_s [S \mapsto r] \qquad$ if there exists S such that $S \xrightarrow{(??\alpha,\vec{\ell},Sys)}_c f$

SpAct $\quad (\alpha, r).S \xrightarrow{(\alpha,\emptyset,Sys)}_s [S \mapsto r]$

(a) Primitive rules

BrCombo $\quad \dfrac{S \xrightarrow{(!\alpha,\vec{\ell},Sys)}_s f \quad P \xrightarrow{(?\alpha,\vec{\ell},Sys)}_c g}{S \parallel P \xrightarrow{(!\alpha?,\vec{\ell},Sys)} f \otimes g} \qquad (f \otimes g)(s) = \begin{cases} f(S')g(P') & \text{if } s \equiv S' \parallel P' \\ 0 & \text{otherwise} \end{cases}$

UniPair $\quad \dfrac{S \xrightarrow{(!!\alpha,\vec{\ell},Sys)}_s f \quad P \xrightarrow{(??\alpha,\vec{\ell},Sys)}_c g}{S \parallel P \xrightarrow{(!!\alpha,\vec{\ell},Sys)}_s f \otimes g} \qquad (f \otimes g)(s) = \begin{cases} f(S')g(P') & \text{if } s \equiv S' \parallel P' \\ 0 & \text{otherwise} \end{cases}$

(b) Combining with capabilities

Parallel $\quad \dfrac{P_1 \xrightarrow{\lambda_s}_s f}{P_1 \parallel P_2 \xrightarrow{\lambda_s}_s f \otimes Id} \qquad (f \otimes Id)(s) = \begin{cases} f(P_1') & \text{if } s \equiv P_1' \parallel P_2 \\ 0 & \text{otherwise} \end{cases}$

$\dfrac{P_2 \xrightarrow{\lambda_s}_s f}{P_1 \parallel P_2 \xrightarrow{\lambda_s}_s Id \otimes f} \qquad (Id \otimes f)(s) = \begin{cases} f(P_2') & \text{if } s \equiv P_1 \parallel P_2' \\ 0 & \text{otherwise} \end{cases}$

Choice $\quad \dfrac{P_1 \xrightarrow{\lambda_s}_s f}{P_1 + P_2 \xrightarrow{\lambda_s}_s f} \qquad \dfrac{P_2 \xrightarrow{\lambda_s}_s f}{P_1 + P_2 \xrightarrow{\lambda_s}_s f} \qquad$ Constant $\quad \dfrac{P \xrightarrow{\lambda_s}_s f \qquad X := P}{X \xrightarrow{\lambda_s}_s f}$

(c) Rules for composition

Figure 2: Stochastic rules for rates

Suppose we want to derive a CTMC for the evolution of the model component *Sys*. For that we need to consider all enabled stochastic transition rules from *Sys*. The CTMC has a transition from the state *Sys* to Sys' if there is a transition $Sys \xrightarrow{\lambda_s}_s f$ such that $f(Sys) \neq 0$. The next step is to consider all transitions from Sys' and so on recursively until no new states are discovered and the full CTMC is generated.

4 Equivalence relations

Firstly we will briefly cover a naive attempt to define a bisimulation on sequential components of PALOMA to demonstrate why it is not entirely trivial to deal with spatial properties of PALOMA models. The approach that allows us to relax the conditions on spatial properties of defined models will be described in more detail.

In terms of semantic rules introduced in Section 3 we are going to say that $S \xrightarrow{a} S'$ holds if there is a stochastic transition $Sys \xrightarrow{\lambda_s}_s f$ and a system Sys' such that $S' \in \mathrm{seq}(Sys')$ and $f(Sys') \neq 0$. In addition the label λ_s is required to be such that

$$\lambda_s = \begin{cases} (\alpha, \emptyset, Sys) & \text{if } a = \alpha \\ (!\alpha, \overrightarrow{\ell}, Sys) & \text{if } a = ?\alpha \vee !\alpha \\ (!!\alpha, \overrightarrow{\ell}, Sys) & \text{if } a = ??\alpha \vee !!\alpha \end{cases}$$

As the behaviour of the PALOMA sequential component is parametrised by its location the natural interpretation would be to consider locations as an inherent part of a component's state. This would lead to the following definition, making use of the syntax-derived rate function defined in Section 2.

Definition 4.1. Let $Sys \in \mathscr{C}$ be any model component serving as a context. A binary relation $\mathscr{R}_{Sys}$ is a bisimulation over sequential components if, and only if, $(S(\ell_1), T(\ell_2)) \in \mathscr{R}_{Sys}$ implies, for all $a \in Act$

1. $R_a(Sys, S(\ell_1)) = R_a(Sys, T(\ell_2))$.
2. $\ell_1 = \ell_2$.
3. $S(\ell_1) \xrightarrow{a} S'(\ell_1')$ implies for some $T'(\ell_2')$, $T(\ell_2) \xrightarrow{a} T'(\ell_2')$ and $(S'(\ell_1'), T'(\ell_2')) \in \mathscr{R}_{Sys}$.
4. $T(\ell_2) \xrightarrow{a} T'(\ell_2')$ implies for some $S'(\ell_1')$, $S(\ell_1) \xrightarrow{a} S'(\ell_1')$ and $(S'(\ell_1'), T'(\ell_2')) \in \mathscr{R}_{Sys}$.

This definition would give rise to an equivalence relation on PALOMA components with respect to the underlying context system. However, Definition 4.1 has some limitations due to the restrictive way in which location is treated, and we will not pursue it further. Specifically, two sequential components which have identical behaviour in different locations will be considered non-equivalent in this setting. This would lead to a very strict equivalence being defined on the model components of PALOMA. A more interesting idea is to shift to considering relative locations between the sequential components. This will be explored in the following subsection.

4.1 Relative locations

In order to consider relative locations between sequential components we need a notion of distance between the components. Thus we consider the case where Loc denotes a metric space. Specifically we will consider the Euclidean plane $\mathbb{R}^2$ (extensions to different metric spaces are immediate).

The notion we make use of in the following discussion is that of isometries – that is, maps between metric spaces that preserve the distances between points. In particular we are interested in the set of Euclidean plane isometries of which we have four types: translations, rotations, reflections and glide reflections. Denote the set of Euclidean plane isometries by $E(2)$.

The first definition we are going to give mimics the Definition 4.1 but allows the locations of the sequential components under consideration to differ by an element in $E(2)$.

Definition 4.2. Let $\phi \in E(2)$ and $Sys \in \mathscr{C}$ a system component serving as context. A binary relation $\mathscr{R}_{\phi,Sys}$ is a bisimulation with respect to ϕ over components if, and only if, $(S(\ell_1),T(\ell_2)) \in \mathscr{R}_{\phi,Sys}$ implies, for all $a \in Act$, that

1. $R_a(Sys,S(\ell_1)) = R_a(Sys,T(\ell_2))$.
2. $\phi(\ell_1) = \ell_2$.
3. $S(\ell_1) \xrightarrow{a} S'(\ell_1')$ implies for some $T'(\ell_2')$, $T(\ell_2) \xrightarrow{a} T'(\ell_2')$ and $(S'(\ell_1'),T'(\ell_2')) \in \mathscr{R}_{\phi,Sys}$.
4. $T(\ell_2) \xrightarrow{a} T'(\ell_2')$ implies for some $S'(\ell_1')$, $T(\ell_2) \xrightarrow{a} S'(\ell_1')$ and $(S'(\ell_1'),T'(\ell_2')) \in \mathscr{R}_{\phi,Sys}$.

In this definition for sequential components the location plays little role. The situation becomes more interesting when we attempt to extend the definition to model components $\mathscr{C}$ of PALOMA.

Definition 4.3. Let $\phi \in E(2)$ and $Sys \in \mathscr{C}_S$ a model component serving as context. A binary relation $\mathscr{R}_{\phi,Sys}$ is a bisimulation with respect to ϕ over model components if, and only if, $(P,Q) \in \mathscr{R}_{\phi,Sys}$ implies, for all $a \in Act$ and all sets of locations L

1. $R_a(L,Sys,P) = R_a(\phi(L),Sys,Q)$.
2. $P \xrightarrow{a} P'$ implies for some Q, $Q \xrightarrow{a} Q'$ and $(P',Q') \in \mathscr{R}_{\phi,Sys}$.
3. $Q \xrightarrow{a} Q'$ implies for some P', $P \xrightarrow{a} P'$ and $(P',Q') \in \mathscr{R}_{\phi,Sys}$.

From the definition we can easily see that any component is bisimilar to itself and that conditions are symmetric – meaning we have $(P,Q) \in \mathscr{R}_{\phi,Sys} \implies (Q,P) \in \mathscr{R}_{\phi,Sys}$ – and that transitivity holds. To be able to define a bisimilarity as the largest bisimulation over the components would require us to verify that a union of bisimulations is again a bisimulation.

Definition 4.4. Two model components P_1,P_2, defined over $\mathbb{R}^2$ are considered bisimilar with respect to context system *Sys*, denoted $P_1 \sim_{Sys} P_2$ if there exists an isometry $\phi \in E(2)$ and a corresponding bisimulation $\mathscr{R}_{\phi,Sys}$ such that $(P_1,P_2) \in \mathscr{R}_{\phi,Sys}$.

The simplest case we can consider is bisimilarity with respect to empty context system *Sys* denoted by $\emptyset$. We illustrate this in the following example.

Example 4.1.

$$
\begin{aligned}
&Transmitter(\ell_0) := !!(message\_move,r)@\mathbf{IR}\{all\}.Transmitter(\ell_1)\\
&Transmitter(\ell_1) := !!(message\_move,r)@\mathbf{IR}\{all\}.Transmitter(\ell_0)\\
&Receiver(\ell_1) \quad := ??(message\_move,p)@\mathbf{Wt}\{v\}.Receiver(\ell_0)\\
&Receiver(\ell_0) \quad := ??(message\_move,q)@\mathbf{Wt}\{v\}.Receiver(\ell_1)
\end{aligned}
$$

For this example take $\ell_0 = (-1,0)$ and $\ell_1 = (1,0)$. The two systems we are going to analyse are

$$
\begin{aligned}
Scenario_1 &:= Transmitter(\ell_0) \,\|\, Receiver(\ell_1)\\
Scenario_2 &:= Transmitter(\ell_1) \,\|\, Receiver(\ell_0)
\end{aligned}
$$

It is clear that the systems are symmetric in the sense that if the locations in $Scenario_1$ are reflected along the y-axis we get $Scenario_2$. Denote the reflection along the y-axis as ϕ. This give $\phi(\ell_0) = \ell_1$ and $\phi(\ell_1) = \ell_0$.

It it intuitively clear that the two systems behave in the same way up to the starting location of the *Transmitter* and *Receiver* in both systems. Thus it makes sense to abstract away the absolute locations and consider the given systems observationally equivalent up to spatial transformation ϕ. In the following we verify that applying Definition 4.3 to these examples indeed agrees with the intuition. The two systems are considered on their own with no additional context – that is the *Sys* in Definiton 4.3 becomes $\emptyset$.

$$R_{!!message\_move}(\ell_0, \emptyset, Scenario_1) = r \qquad R_{??message\_move}(\ell_1, \emptyset, Scenario_2) = rp$$

and

$$\begin{aligned} R_{!!message\_move}(\phi(\ell_0), \emptyset, Scenario_1) = R_{!!message\_move}(\ell_1, \emptyset, Scenario_1) = r \\ R_{??message\_move}(\phi(\ell_1), \emptyset, Scenario_2) = R_{!!message\_move}(\ell_0, \emptyset, Scenario_2) = rp \end{aligned}$$

As the rest of the rates are 0 then the first condition in Definition 4.3 holds. To get the second and third conditions requires verifying that the rates also match for derivatives of the systems $Scenario_1$ and $Scenario_2$. This is not going to be done here but one can easily see that the same symmetries are going to hold throughout the evolution of the systems and thus the Definition 4.4 would give that

$$Scenario_1 \sim_{\emptyset} Scenario_2$$

In the example we gave no additional context to the systems under study but the Definition 4.3 allows for reasoning about the equivalence of the two components in the context of any given system. The following example demonstrates that components being equivalent with respect to one context system does not imply equivalence with respect to other contexts.

Example 4.2.

$$\begin{aligned} &Transmitter(\ell_0) &&:= !!(message, r)@\mathbf{IR}\{all\}.Transmitter(\ell_0) \\ &Receiver(\ell_0) &&:= ??(message, r)@\mathbf{IR}\{all\}.Receiver(\ell_0) \\ &Sys &&:= Transmitter(\ell_0) \end{aligned}$$

It can be verified that according to Definition 4.3 we have

$$Transmitter \sim_{\emptyset} Receiver$$

as neither component can perform an action due to the blocking nature of unicast communication. On the other hand we have

$$Transmitter \not\sim_{Sys} Receiver$$

as the system $Transmitter(\ell_0) \parallel Transmitter(\ell_0)$ would not perform an action while in $Transmitter(\ell_0) \parallel Receiver(\ell_0)$ we have unicast communication happening.

5 Conclusions and Future Work

The paper introducing the PALOMA language in its current form [**?**] concentrated on the fluid analysis of CTMCs defined on population counts and gave semantic rules for generating a model in the Multi-message Multi-class Markovian Agents Model framework [**?**]. In order to have a rigorous foundation for bisimulation definitions we have introduced the new agent level semantics in the FuTSs framework [**?**]. Several other process algebras that capture the relative locations of interacting entities have been developed. In relation to systems biology there is, for example, SpacePi [**?**] where locations are defined in real coordinate spaces and for wireless networks there is, for example, CWS [**?**] which makes no restrictions on the notion of location that can be used. However, there is very little work exploring notions of equivalence for spatially distributed systems.

We presented an idea for a bisimulation of PALOMA models which allows us to abstract away explicitly defined locations of PALOMA components and use relative locations of sequential components as the basis of the model comparison. This idea relies on working over the Euclidean plane and being able to apply isometries to the model components of PALOMA leaving the relative spatial structure of the model components intact. As the behaviour of PALOMA components is dependent on the context in which they appear thus definitions of equivalences are given in terms of the context system.

The bisimulation ideas presented are intended to serve as a grounding for further development of model comparison and analysis methods for systems with explicitly defined spatial location. From the modelling and simulation perspective the aim of equivalence relations is to provide formal ways of reducing the state space of the underlying CTMC by allowing us to swap out components in the model for ones generating a smaller state space while leaving the behaviour of the model the same up to some equivalence relation. In particular, it is useful to consider such equivalence relations that induce a lumpable partition at the CTMC level.

Acknowledgements

This work was supported by grant EP/L01503X/1 for the University of Edinburgh School of Informatics Centre for Doctoral Training in Pervasive Parallelism (http://pervasiveparallelism.inf.ed.ac.uk/) from the EPSRC, and by the EU-funded project, QUANTICOL 600708.